



Customizing Form Guides Using Flex™ Builder™

July 2007

Adobe® LiveCycle™ ES

Version 8.0

© 2007 Adobe Systems Incorporated. All rights reserved.

Adobe® LiveCycle® ES 8.0 Customizing Form Guides Using Flex™ Builder™ for Microsoft® Windows®

Edition 1.0, July 2007

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names, company logos and user names in sample material or sample forms included in this documentation and/or software are for demonstration purposes only and are not intended to refer to any actual organization or persons.

Adobe, the Adobe logo, Acrobat, Flex, Flex Builder, LiveCycle, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This product contains either BISAFE and/or TIPEM software by RSA Data Security, Inc.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes code licensed from RSA Data Security.

This product includes software developed by the JDOM Project (<http://www.jdom.org/>).

Macromedia Flash 8 video is powered by On2 TrueMotion video technology. © 1992-2005 On2 Technologies, Inc. All Rights Reserved.
<http://www.on2.com>.

This product includes software developed by the OpenSymphony Group (<http://www.opensymphony.com/>).

Portions of this code are licensed from Nellymoser(www.nellymoser.com).

MPEG Layer-3 audio compression technology licensed by Fraunhofer IIS and THOMSON Multimedia (<http://www.iis.fhg.de/amm/>).

This product includes software developed by L2FProd.com (<http://www.L2FProd.com/>).

The JBoss library is licensed under the GNU Library General Public License, a copy of which is included with this software.

The BeanShell library is licensed under the GNU Library General Public License, a copy of which is included with this software.

This product includes software developed by The Werken Company.

This product includes software developed by the IronSmith Project (<http://www.ironsmith.org/>).

The OpenOffice.org library is licensed under the GNU Library General Public License, a copy of which is included with this software.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

About This Document.....	4
Who should read this document?	4
Additional information.....	4
1 About Customizing Form Guides Using Flex Builder.....	6
2 Creating Flex Library Projects for Custom Form Guides	7
Flex library project folder structure	8
Building your custom Flex library project	8
Importing sample Flex library projects.....	9
What's next?	9
3 Creating Form Guide Layouts	10
Creating custom form guide layouts	11
Form guide layout sample	12
What's next?	13
4 Creating Panel Layouts	14
Panel layout sample	15
What's next?	16
5 Creating Controls.....	17
Next panel button example.....	18
Tree navigation example.....	18
What's next?	19
6 Custom Component Lists	20
Button components	21
Help components.....	21
Label components	22
Navigation components	22
Output components.....	22
7 Form Guide CSS Styles	23
Form guide CSS classes	23
Form guide CSS properties.....	34
Index	51

About This Document

Welcome to Customizing Form Guides Using Flex™ Builder™. This document provides information about creating custom form guide layouts, panels layouts, and controls using Adobe® Flex Builder.

Note: You must use either Flex Builder 2.0.1 Hot Fix 2 or higher, or Flex SDK 2.0.1 Hot Fix 2 or higher.

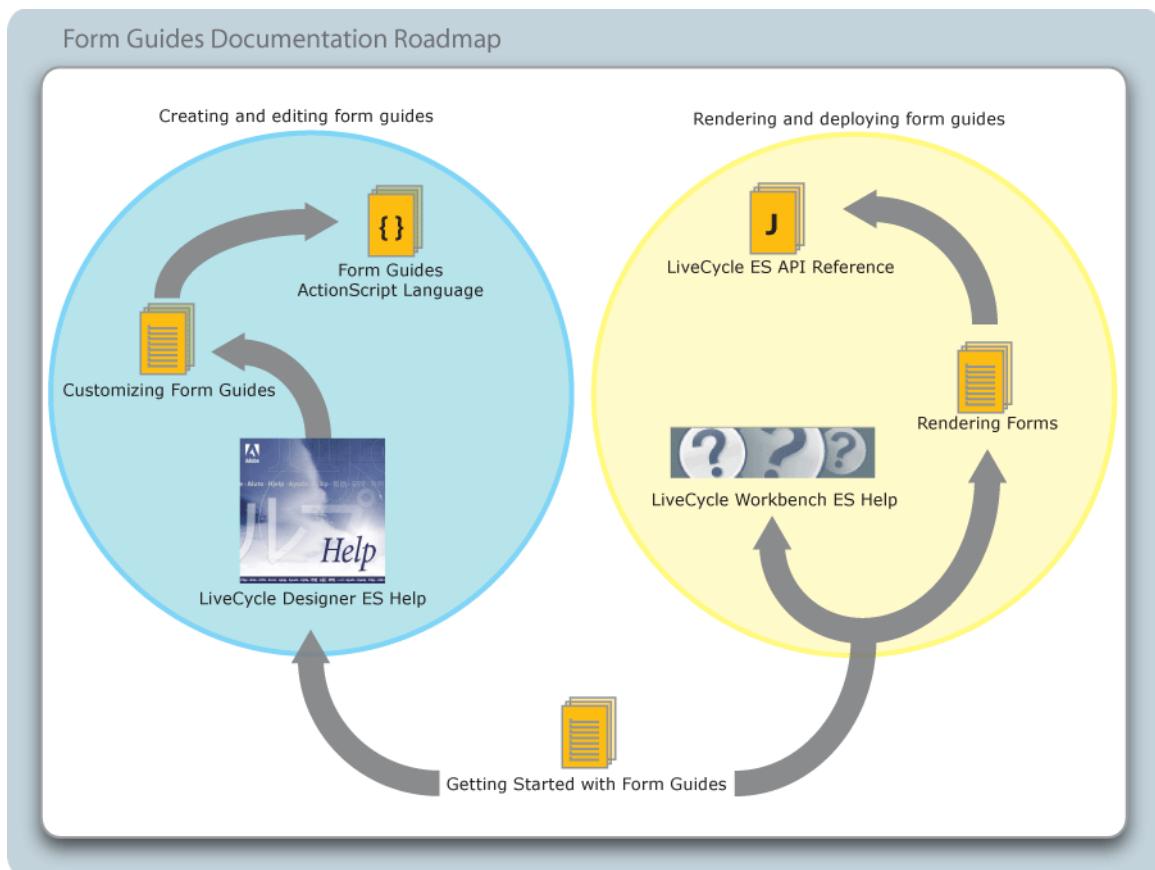
Who should read this document?

This document is intended for Flex developers who are interested in learning about how to create custom form guide components to extend existing form guide components shipped with Guide Builder, or to create new form guide components to meet specific needs.

A knowledge of form guides, Guide Builder, and Flex Builder is assumed.

Additional information

Adobe has a wide variety of resources about form guides focused at a variety of audiences. The following illustration and section outline the various resources available.



To view these resources, go to the location specified in the See column in the following table.

For information about	See
Detailed information about creating and editing form guides using Guide Builder.	<i>LiveCycle Designer ES Help</i> , available within LiveCycle Designer ES and also available at http://www.adobe.com/go/learn_lc_designer
The starting point for learning about form guides. This guide includes a walkthrough of creating a form guide from an existing form design and an example of how to render and deploy a form guide to Adobe LiveCycle Workspace ES by using processes created in LiveCycle Workbench ES.	<i>Getting Started with Form Guides</i> , available at http://www.adobe.com/go/learn_lc_fgGetStart
The ActionScript™ classes and properties included with LiveCycle ES (Enterprise Suite).	<i>LiveCycle ES ActionScript Language</i> , available at www.adobe.com/go/learn_lc_fgActionScript
Rendering and deploying form guides using processes created in LiveCycle Workbench ES.	<i>LiveCycle Workbench ES Help</i> , available within LiveCycle Workbench ES and also available at http://www.adobe.com/go/learn_lc_workbench
Rendering a form guide using the Adobe LiveCycle LiveCycle Forms ES API.	<i>Developing Applications Using LiveCycle ES APIs</i> , available at www.adobe.com/go/learn_lc_programming
Detailed information about the classes and methods included with LiveCycle ES.	<i>LiveCycle ES Java API Reference</i> , available at www.adobe.com/go/learn_lc_JavaAPI
LiveCycle ES terminology	<i>LiveCycle ES Glossary</i> , available at www.adobe.com/go/learn_lc_glossary
Other services and products that integrate with LiveCycle ES	www.adobe.com
Patch updates, technical notes, and additional information on this product version	www.adobe.com/support/products/enterprise/index.html

Using Flex Builder, you can go beyond the form guide customizing options available in Guide Builder to create custom form guide components designed to suit your specific needs. For example, a Flex developer could create a new custom control that displays form guide sections and panels in a tree structure.

The process for creating new custom form guide components, whether they are form guide layouts, panel layouts, or form guide controls, follows the same general steps:

1. Create a new Flex Library project in Flex Builder. For more information, see [Creating Flex Library Projects for Custom Form Guides](#).
2. Create new customized form guide components as part of the Flex Library project. For more information, see one of the following sections:
 - [Creating Form Guide Layouts](#)
 - [Creating Panel Layouts](#)
 - [Creating Controls](#)
3. Compile the Flex library project to a SWC file. For more information, see [Building your custom Flex library project](#).
4. Import the SWC file into Guide Builder. For more information about importing SWC files into Guide Builder, see *LiveCycle Designer ES Help*.
5. Apply the new form guide components to a form guide.

Creating Flex Library Projects for Custom Form Guides

The first step in creating custom form guide components is to create a new Flex library project in Flex Builder. When completed, the Flex library projects you create must be saved to a SWC file for importing into Guide Builder. A *SWC file* is an archive file for Flex components and other assets.

Each Flex library project you create must include the following elements:

- Locale-specific form guide SWC files.
- Locale-specific form guide SWC resource bundles.
- A folder structure with specific subfolders for each type of form guide component. For more information about creating the correct folder structure, see [Flex library project folder structure](#).

In addition, you can optionally include additional SWC files that contain MXML components or ActionScript classes that you want to leverage, or you can include other assets such as image files or videos.

► To create a custom form guide Flex library project:

1. Start Flex Builder.
2. Select **File > New > Flex Library Project**.
3. Type a project name, assign a workspace, and then click **Next**.
4. Click **Library Path**.
5. Click **Add SWC**.
6. Browse to the \plugins\GuideBuilder folder where Adobe LiveCycle Designer ES is installed. By default, the path is C:\Program Files\Adobe\LiveCycle Designer ES\8.1\plugins\GuideBuilder. Add the following SWC files to the project:
 - GAClientRuntime.swc
 - XFAModel.swc
7. Browse to the \plugins\GuideBuilder\locale folder and add the following SWC files from the appropriate language folder to the project:
 - GAClientRuntime_rb.swc
 - XFAModel_rb.swc
8. Click **Finish**.

After you create the Flex library project, you must create a folder structure to store your custom form guide components. Guide Builder requires that each type of custom form guide component be in a specifically named project folder.

Flex library project folder structure

Each custom Flex library project must include form guide components stored within a specific folder structure. When you compile the project to a SWC file and import the SWC file into Guide Builder, form guide components are loaded from the project folders and made available through the Guide Builder interface.

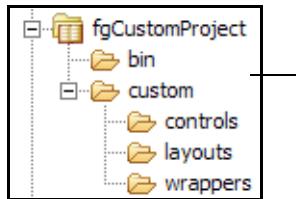
The custom Flex library folder structure must contain the following elements:

- A top-level folder that has a unique name. The top-level folder in the basic folder structure example below is named *custom*.

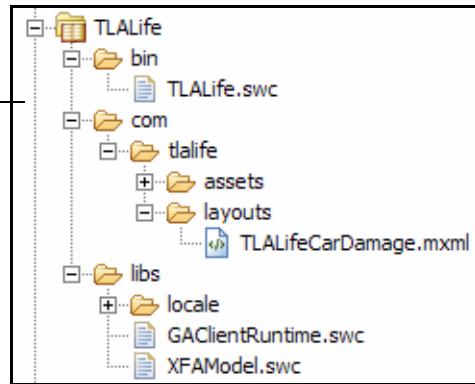
Note: The folder cannot be named *ga*.

- At least one nested subfolder that has one of the following names:
 - wrappers** To contain form guide layouts.
 - layouts** To contain panel layouts.
 - controls** To contain form guide controls, such as panel navigation.

For example, the images below illustrate valid folder structures.



A more common project folder structure that includes one custom panel layout and additional assets, such as images, in the assets folder.



Building your custom Flex library project

After you create custom form guide components, to make them available for use in form guides you must compile the Flex library project to a SWC file and then import the SWC file into Guide Builder.

► To compile your custom Flex library project:

1. Start Flex Builder.

2. If your Flex library project is set to compile automatically, proceed to step 3. If your Flex library project is not set to compile automatically, in the Navigation view, select your project and click **Project > Build Project**.
3. Make the compiled project SWC file available to Guide Builder in one of the following ways:
 - If you are compiling form guides as part of a LiveCycle ES solution and you want to distribute your SWC files by way of the LiveCycle ES repository, open LiveCycle Workbench ES and drag your SWC file to a folder within the repository.
 - Publish your SWC file to a shared network location.

Importing sample Flex library projects

Included with the LiveCycle ES SDK are preconfigured Flex Builder projects to help you start creating custom components more quickly. The LiveCycle ES SDK contains the following projects:

GuideControls Contains copies of the Adobe form guide controls that are included with the Guide Builder SWC files.

GuideLayouts Contains copies of the form guide layouts that are included with Guide Builder.

GuideSource Contains all form guide source ActionScript and MXML files.

PanelLayouts Contains copies of the Adobe panel layouts that are included with Guide Builder.

TLALife A practical sample of a custom Flex library project. The TLALife project is used to create the TLA Life sample form guide that is included with LiveCycle Designer ES, and is used as an example throughout this document.

You can use the Flex Builder projects as a reference when creating your own custom Flex library projects.

Note: You should only copy content from the sample Flex Builder projects, and not use them as a starting point for creating new custom form guide components.

➤ To import the sample form guide Flex Builder projects:

1. Navigate to the LiveCycle ES SDK\samples\FormGuides folder located where you installed LiveCycle ES (by default, C:\Adobe\LiveCycle8).
2. Copy the files to your Flex Builder workspace.
3. Start Flex Builder.
4. Select **File > Import**.
5. Select **Existing Projects Into Workspace** and click **Next**.
6. Ensure that **Select Root Directory** is selected and click **Browse**.
7. Select the folder from which you extracted the contents of the ZIP file and click **Finish**.

What's next?

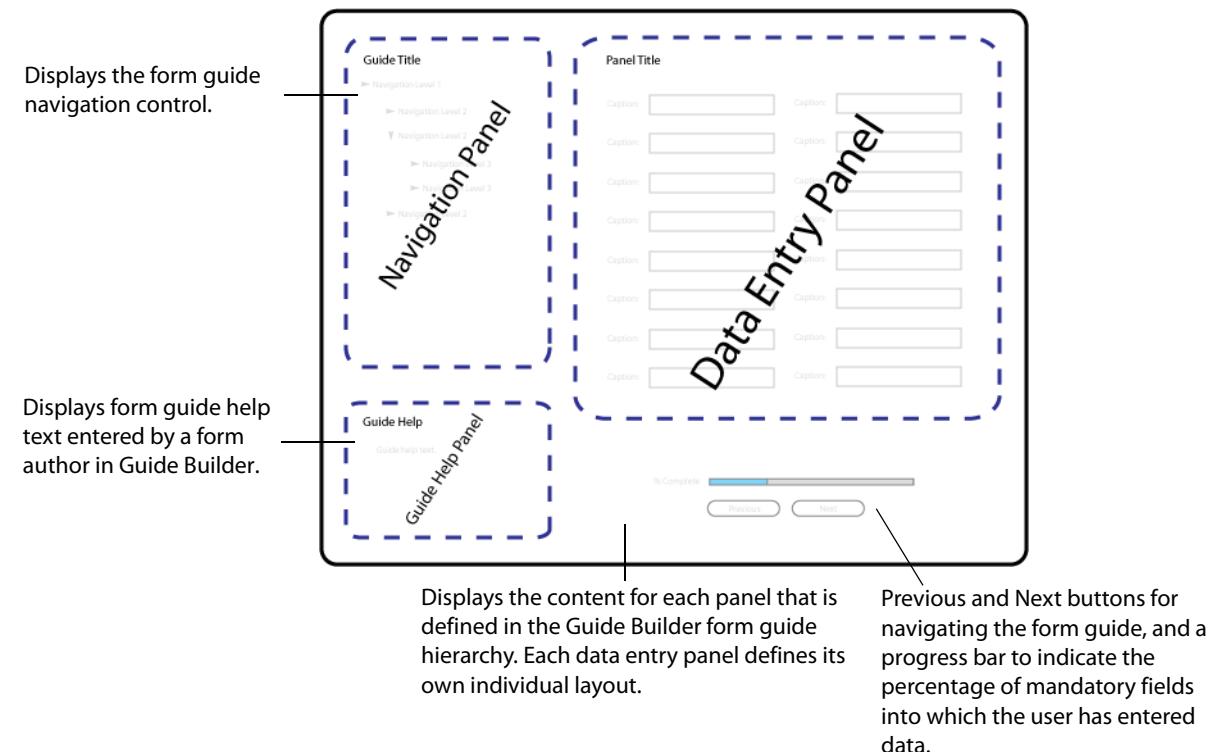
After you have imported the sample form guide Flex Builder projects, you can start creating your own form guide layouts. For information about form guide layouts, see [Creating Form Guide Layouts](#).

Creating Form Guide Layouts

A form guide layout defines the visual layout and structure of a form guide that remains constant throughout a form-filling session. In general, a form guide layout consists of the following components:

- Guide help
- Progress bar
- Previous button
- Next button
- Navigation control
- Panel content

The following image illustrates an example of how a form guide layout is structured.



Creating custom form guide layouts

Follow this procedure to create each custom guide layout.

► **To create a custom form guide layout:**

1. Start Flex Builder.
2. Select **File > New > MXML Component**.
3. Type a unique file name.
4. In the **Based On** list, select **Wrapper**.
5. (Optional) Set values for **Width** and **Height**.
6. Click **Finish**.

After you create the shell of the new form guide layout, you add Flex Builder components, either in Source view or Design view, based on the behavior you are trying to achieve. Typically, a form guide layout includes standard Flex Builder layout containers such as VBox, HBox, Canvas, and Panel. In addition, you can add specific form guide components from the list in the Custom folder within the Components view. The following form guide components are the most common ones that appear in a form guide layout:

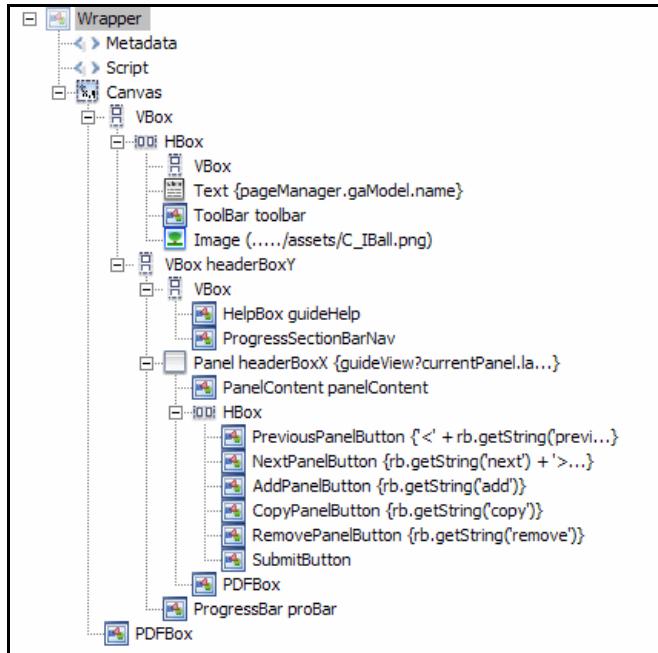
- Navigation control such as AccordionNav, MXTreeNav, ProgressSectionBarNav, or TabTabNav
- ProgressBar
- HelpBox
- ToolBar
- PanelContent
- Guide actions such as PreviousPanelButton, NextPanelButton, AddPanelButton, CopyPanelButton, RemovePanelButton, and SubmitButton
- PDFBox

For more information about these form guide components, see [Custom Component Lists](#).

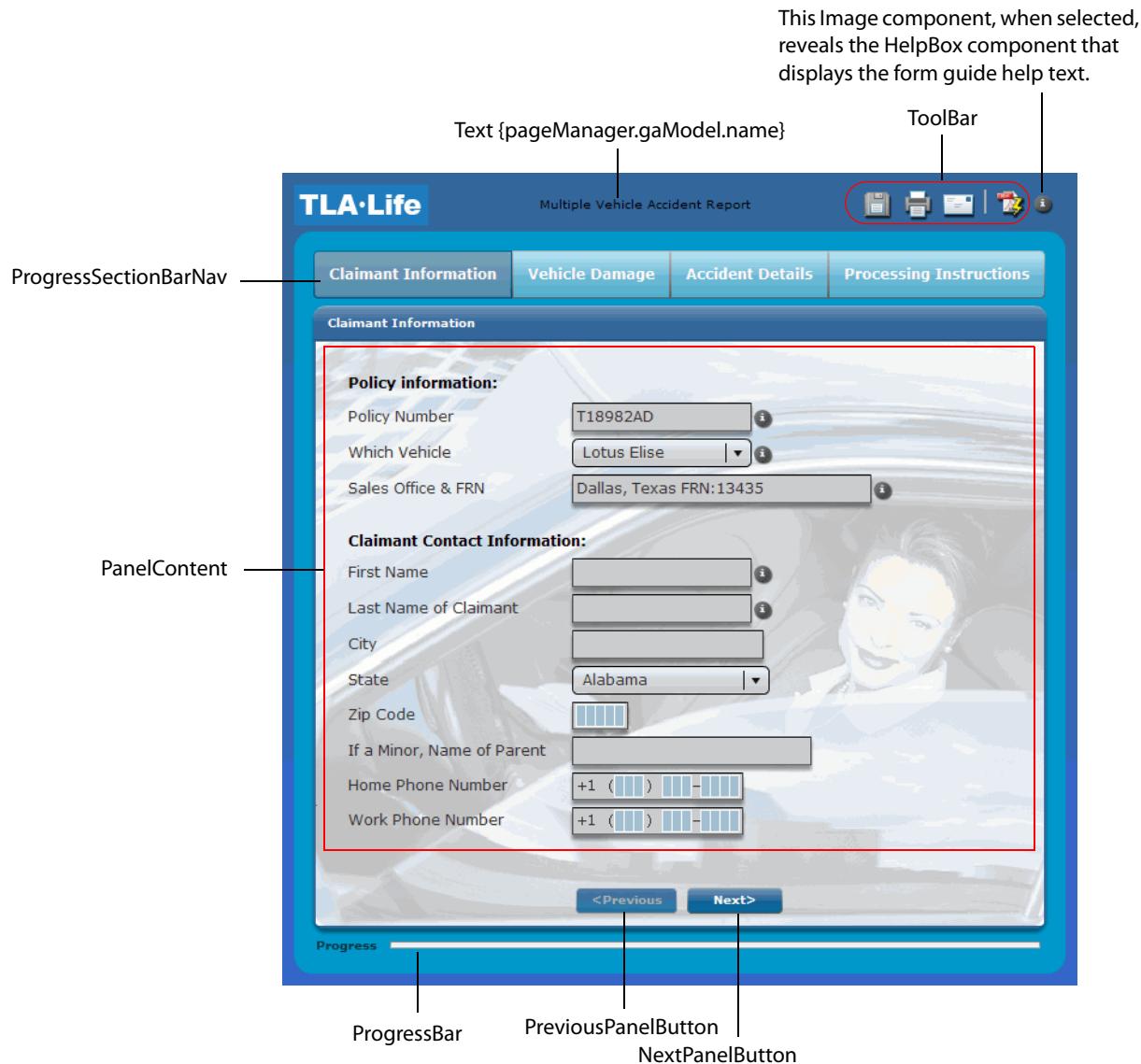
Form guide layout sample

To better understand how to structure form guide layouts, this document will use the TLALife Flex Builder project as an example. Of the sample Flex Builder projects, TLALife is the most representative in terms of project structure, folder naming, and incorporating custom components into a form guide design. The TLALife Flex Builder project was used to create the TLALife form guide included with the TLALife form design. For more information, see [Importing sample Flex library projects](#).

At the form guide layout level, the TLALife project uses the default Button Bar layout included with Guide Builder. The following image illustrates the Outline view of the Button Bar form guide layout.



The Button Bar layout is structured using a series of standard Flex Builder VBox and HBox components combined with custom form guide components. The following image illustrates the output of the TLA Life sample form guide and calls out the various custom form guide components.



What's next?

After you are familiar with how a form guide layout is structured, start with a new Flex library project and create a variation on an existing form guide layout. For example, restructure the Button Bar layout by reordering the HBox containing the logo, form guide title, ToolBar component, and Image component so that it appears at the bottom of the layout instead of at the top.

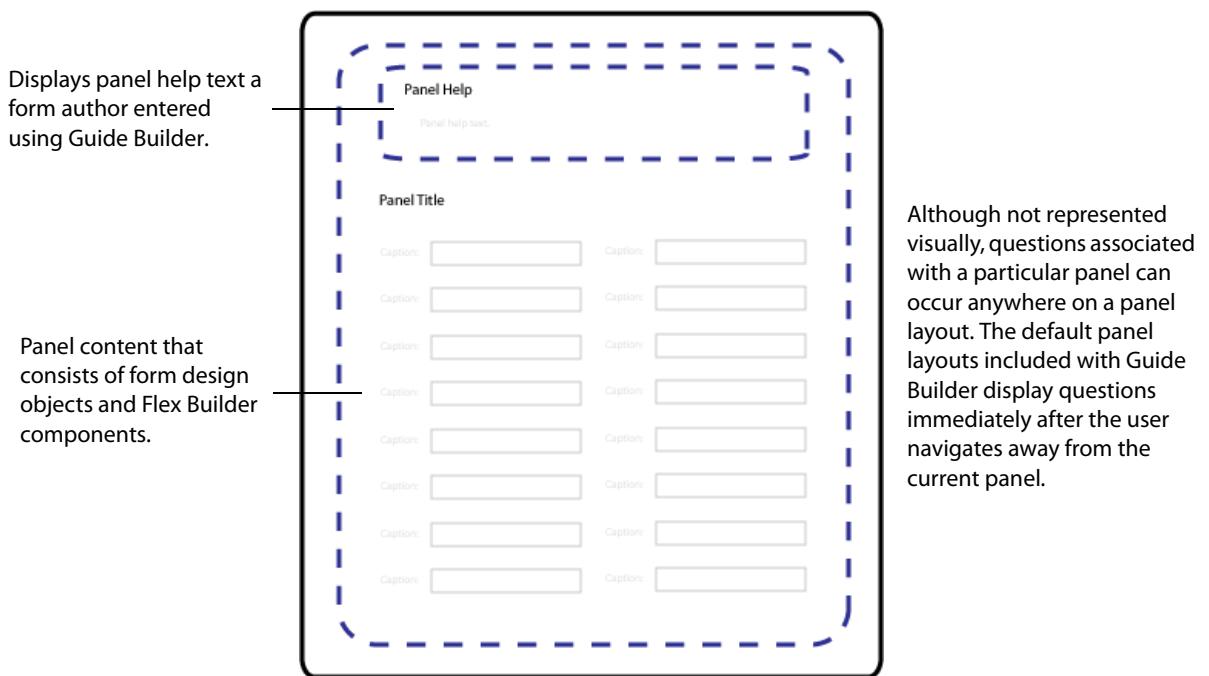
After you create a custom form guide layout, you must compile your Flex library project and then import the SWC file into Guide Builder. After the new layout is imported, it can be applied to form guides.

For information about building your Flex library project, see [Building your custom Flex library project](#). For more information about importing SWC files into Guide Builder, see *LiveCycle Designer ES Help*.

A *panel layout* defines the visual layout and presentation of form design objects associated with a single form guide hierarchy panel. In general, a panel layout consists of the following components:

- Panel help
- Panel content
- Questions to further structure form guide navigation

The following image illustrates an example of how a panel layout is structured.



► **To create a panel layout:**

1. Start Flex Builder.
2. Select **File > New > MXML Component**.
3. Type a unique file name.
4. In the **Based On** list, select **LayoutTemplate**.
5. (Optional) Set values for **Width** and **Height**.
6. Click **Finish**.

After you create the shell of the new panel layout, you add Flex Builder components, either in Source view or Design view, based on the behavior you are trying to achieve. Typically, a panel layout includes standard Flex Builder layout containers such as VBox and HBox. In addition, you might add specific form guide components from the list in the Custom folder within the Components view. The following form guide components are the most common ones that appear in a panel layout:

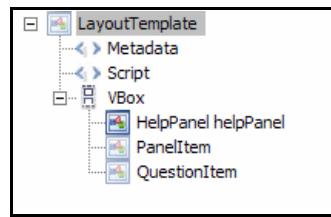
- HelpPanel
- PanellItem
- QuestionItem

However, panel layouts are not required to contain any specific form guide components. For example, a panel may contain hard-coded text objects, such as legal text or instructions, that must be read prior to filling the form guide. The TLALife sample form guide discussed in [Panel layout sample](#) includes a section called Processing Instructions that consists of a panel that contains a single text object.

For more information about form guide components, see [Custom Component Lists](#). For more information about panel layouts and their components, see *LiveCycle ES ActionScript Reference*, available at www.adobe.com/go/learn_lc_fgActionScript.

Panel layout sample

Using the TLALife project as an example, most of the panels in the form guide hierarchy use the One Column panel layout included with Guide Builder. The following image illustrates the Outline view of the One Column panel layout.



The One Column layout is structured as a single VBox that contains the following form guide components in the order shown:

HelpPanel Displays panel help text entered by a form author using Guide Builder.

PanellItem Displays form design objects contained within a panel.

QuestionItem Displays any questions entered by a form author using Guide Builder.

The following image illustrates the output of the first panel in the TLALife sample form guide and calls out the various form guide components within the panel.

By default, the panel does not include panel help text; therefore, no help text is displayed in the output.

Policy information:

Policy Number: T18982AD

Which Vehicle: Lotus Elise

Sales Office & FRN: Dallas, Texas FRN:13435

Claimant Contact Information:

First Name

Last Name of Claimant

City

State: Alabama

Zip Code

If a Minor, Name of Parent

Home Phone Number: +1 () -

Work Phone Number: +1 () -

Each text object and field contained within the panel in the Guide Builder form guide hierarchy is displayed vertically as a series of PanelItem components.

When a form author includes answer-based navigation for this panel, any questions will be displayed when the end user navigates away from the current panel, either by using the Previous/Next buttons or by selecting another entry in the form guide navigation panel.

What's next?

After you are familiar with how a panel layout is structured, start with a new Flex library project and create a variation of an existing panel layout. For example, restructure the One Column layout by reordering the QuestionItem component so that it occurs before the HelpPanel component.

After you create a custom panel layout, you must compile your Flex library project and then import the SWC file into Guide Builder. After the new layout is imported, it can be applied to form guides.

For information about building your Flex library project, see [Building your custom Flex library project](#). For more information about importing SWC files into Guide Builder, see [LiveCycle Designer ES Help](#).

Controls are form guide components that communicate directly with a form guide to perform specific actions. For example, controls allow users to navigate through a form guide, add and remove panels, and display panel and section titles. In general, a control performs only a single action.

To take full advantage of custom controls within a form guide, you should be familiar with the form guide ActionScript packages. For more information, see the *LiveCycle ES ActionScript Reference* available at www.adobe.com/go/learn_lc_fgActionScript.

► **To create a new custom MXML control:**

1. Start Flex Builder.
2. Select **File > New > MXML Component**.
3. Type a unique file name.
4. In the **Based On** list, select a component that is the most similar to the component you want to create, in terms of behavior. This allows you to take advantage of existing behaviors through inheritance.
5. (Optional) Set values for **Width** and **Height**.
6. Click **Finish**.

► **To create a new custom ActionScript control:**

1. Start Flex Builder.
2. Select **File > New > ActionScript Class**.
3. Type a unique class name.
4. In the **Superclass** field, specify the ActionScript class that is the most similar to the component you want to create, in terms of behavior. This allows you to take advantage of existing behaviors through inheritance.
5. (Optional) Set values for **Width** and **Height**.
6. Click **Finish**.

After you create the shell of the new control, you add functionality using the Source view in Flex Builder based on the behavior you are trying to achieve. Most controls will require you to add ActionScript that takes advantage of the form guide ActionScript packages included with the form guide SWC files and resource bundles. For more information, see the *LiveCycle ES ActionScript Reference* available at www.adobe.com/go/learn_lc_fgActionScript.

Next panel button example

A simple example of a form guide control is a button that navigates an end user forward one panel within a form guide. In this case, the button should behave according to the following rules:

- Disable if the current panel is the last panel in the form guide.
- Move to the next panel as defined by the Guide Builder form guide hierarchy.

The following ActionScript is an example of a next panel button control that extends the default Flex button control.

Example: *Next panel button*

```
<?xml version="1.0" encoding="utf-8"?>

<!-- Specifies that the button should only be enabled if the current panel is
not the last panel, and if a user clicks the button the form guide retrieves
the next panel in the hierarchy. -->

<mx:Button click="{panels.selectedIndex++}"
    enabled="{panels.selectedIndex < panels.length-1}"
    xmlns:mx="http://www.adobe.com/2006/mxml">

<!-- Gets the panels ActionScript class instance that includes the list of
visible the panels in a form guide. -->
<mx:Script>
    <![CDATA[
        import ga.views.PanelListView;
        private var panels:PanelListView = PanelListView.instance;
    ]]>
</mx:Script>
</mx:Button>
```

Tree navigation example

Another example of a form guide control is a tree navigation control that displays a list of the sections and panels within a form guide. The tree navigation control button should demonstrate the following behavior:

- When a user selects a section in the tree, the section expands to display the panels within the section. In addition, the first panel in the section is displayed in the area of the form guide designated to display panel content.
- When a user selects a panel in the tree, that panel is displayed in the area of the form guide designated to display panel content.

The following ActionScript is an example of a tree navigation control that uses the default Flex Builder `mx:Tree` class.

Example: *Tree navigation*

```
<?xml version="1.0" encoding="utf-8"?>

<!-- Information about the form guide sections and panels is passed to the
mx:Tree object as a newdataProvider. The currently selected item in the tree
is always the currently selected section or panel within the form guide. -->

<mx:Tree dataProvider="{treeView}"
          selectedItem="{treeView.selectedItem}"
          change="{treeView.selectedItem = this.selectedItem}"
          xmlns:mx="http://www.adobe.com/2006/mxml">

<!-- References the TreeView ActionScript class to create a new TreeView
instance that includes information on the sections and panels within a form
guide. -->

<mx:Script>
<! [CDATA[
    import ga.views.TreeView;
    private var treeView:TreeView = TreeView.instance;
]]>
</mx:Script>
</mx:Tree>
```

What's next?

After you create custom components, you must include them in custom form guide layouts or panel layouts.

For information about creating custom form guide layouts and panel layouts, see [Creating Form Guide Layouts](#) and [Creating Panel Layouts](#).

When you create a custom Flex library project, you must include the form guide SWC files and resource bundles specific to your locale. Including the SWC files and resource bundles adds a collection of custom form guide components to the Custom folder in your Flex Builder Components view. The custom form guide components include everything you need to create custom form guide layouts and panel layouts.

Note: When you are in Design view, you cannot add all of the custom form guide components that appear in the Custom folder to form guide layouts or panel layouts. Components that you can add are indicated with an active component icon , and those you cannot add are indicated with an inactive component icon . Additionally, custom components that you create appear in the Custom folder with a new component icon .

The custom components that you can add to layouts are listed alphabetically in the Flex Builder Components view but conceptually fall into the following categories:

- [Button components](#)
- [Help components](#)
- [Label components](#)
- [Navigation components](#)
- [Output components](#)

Button components

Button components provide the most common form guide actions.

AddPanelButton Adds a new panel to a list of repeating panels. This button is available only when the following statements are true:

- The current panel can repeat.
- Adding a new panel does not conflict with the maximum occurrence value of the associated subform object on the form.

CopyPanelButton Creates a copy of the currently selected panel and adds it to the list of repeating panels. This button is available only when the following statements are true:

- The current panel can repeat.
- Adding a copy of the current panel does not conflict with the maximum occurrence value of the associated subform object on the form.

NextPanelButton Displays the next panel in the form guide hierarchy. If the current panel is the last panel in the hierarchy, this button is not available.

PreviousPanelButton Displays the previous panel in the form guide hierarchy. If the current panel is the first panel in the hierarchy, this button is not available.

RemovePanelButton Removes the current panel from the list of repeating panels. This button is available only when the following statements are true:

- The current panel can repeat.
- Removing the current panel does not conflict with the minimum occurrence value of the associated subform object on the form.

SubmitButton Displays a submit button, but only when the current panel is the last panel in the form guide hierarchy. When clicked, this button performs one of the following actions, depending on the values selected in the Submit From list in Guide Builder:

PDF Submits the form guide data from the PDF form. Clicking the submit button in the form guide opens the PDF form, from which the form filler can submit the data. If the PDF form is not available, clicking the submit button in the form guide instructs LiveCycle Forms ES to create the PDF form and return it to the browser so that the form filler can verify it and then click the Submit button in it.

Printed Form Clicking the submit button opens the PDF form so that the form filler can print the form.

Guide Clicking the submit button submits the data from the form guide.

Hosted Application Specifies that the hosted application, such as LiveCycle Workspace ES, controls the data submission. In this case, the form guide does not have a submit button. The hosted application extracts the data from the form guide and performs the data submission.

Help components

Help components provide the ability to display help to end users in either text or video format:

HelpBox Displays form guide help, either text or video.

HelpPanel Displays panel help, either text or video.

HelpVideo Displays the help video control.

Label components

Label components provide objects that display section and panel titles:

PanelTitle A label that displays the name of the currently selected panel.

SectionTitle A label that displays the name of the currently selected section.

Navigation components

Navigation components provide the system for navigating the sections and panels of a form guide:

AccordionNav An accordion menu composed of sections that each contain a list of panels. The default Guide Builder form guide layout named *Left Accordion* is an example of the usage of the AccordionNav component.

MxTreeNav A tree structure that lists multiple section and panel levels. The default Guide Builder form guide layout named *Left Multilevel* is an example of the usage of the MxTreeNav component.

Note: The MxTreeNav component is the only navigation component that displays nested section and panel levels.

ProgressSectionBarNav A horizontal list of buttons that represents each section in the form guide hierarchy. The default Guide Builder form guide layout named *Button Bar* is an example of the usage of the ProgressSectionBarNav component.

Note: This navigator is useful when each form guide section includes only one panel.

TabTabNav A navigation system that consists of two corresponding levels of tab menus. The top-level tabs list the sections in the form guide hierarchy and the bottom-level tabs list the panels for the currently selected top-level tab. The default Guide Builder form guide layout named *Workspace* is an example of the usage of the TabTabNav component.

Output components

Output components provide objects that display content or functionality to users at run time.

PanelContent Displays the content of form guide panels.

PDFBox Contains the PDF view of the form if the PDF is rendered with the form guide.

ProgressBar Displays the percentage of mandatory fields into which an end user has entered data. This control is not available if the form guide does not contain mandatory fields.

ToolBar Displays the form guide toolbar, which includes the Save PDF, Print PDF, Email PDF, and Show/Hide PDF controls.

This section lists the form guide CSS styles recognized by Guide Builder. You can use this style information to create a common CSS file that you can share among multiple form guides to maintain common styling.

Form guide CSS classes

.application

The `.application` class specifies overall style properties for a form guide application.

Property	Default value
backgroundAlpha	1.0
backgroundColor	#336699
backgroundGradientColors	#336699, #3366CC
basewrapper	ga.wrappers.LeftAccordion
paddingBottom	10
paddingLeft	10
paddingRight	10
paddingTop	5
version	1.0.0

.buttons

The `.buttons` class defines the style properties for button objects.

Property	Default value
alpha	1.0
backgroundAlpha	1.0
color	#FFFFFF
fillAlphas	1.0,1.0
fillColors	#336699, #006699
themeColor	#336699

.fieldhelp

The `.fieldhelp` class specifies properties for field level help on a form guide.

Property	Default value
<code>backgroundColor</code>	#666666
<code>borderColor</code>	#666666
<code>color</code>	#FFFFFF
<code>dropShadowEnabled</code>	true
<code>gradientColors</code>	#333333, #666666
<code>fontSize</code>	12
<code>fontStyle</code>	normal
<code>fontWeight</code>	normal
<code>textDecoration</code>	normal

.guide

The `.guide` class specifies style properties for the top level layer of the form guide application as defined in the Guide Wrapper/Layout definition.

Property	Default value
<code>backgroundAlpha</code>	1.0
<code>backgroundColor</code>	#0099CC
<code>barColor</code>	#3366cc
<code>borderAlpha</code>	0.7
<code>borderColor</code>	#0099CC
<code>borderStyle</code>	solid
<code>cornerRadius</code>	15
<code>headerHeight</code>	56
<code>paddingBottom</code>	15
<code>paddingLeft</code>	15
<code>paddingRight</code>	15
<code>paddingTop</code>	15

.guidehelp

The `.guidehelp` class specifies properties for guide level help on a form guide.

Property	Default value
alpha	1.0
backgroundAlpha	1.0
backgroundColor	#FFFFFF
borderColor	#336699
color	#FFFFFF
cornerRadius	10
headerAlphas	1.0,1.0
headerColors	#336699, #336699
roundedBottomCorners	true
textDecoration	normal

.layoutobjects

The `.layoutobjects` class defines the margins around an object within a panel layout.

Property	Default value
paddingBottom	
paddingLeft	
paddingRight	
paddingTop	

.layoutrepeaterobjects

The `.layoutrepeaterobjects` class defines the margins around an object within a repeating panel layout.

Property	Default value
paddingBottom	20
paddingLeft	25
paddingRight	25
paddingTop	25

.logo

The `.logo` class controls the space reserved for a logo graphic.

Property	Default value
paddingBottom	0
paddingRight	0

.navigationbase

The `.navigationbase` class specifies that area that surrounds the navigation buttons.

Property	Default value
horizontalAlign	center
paddingBottom	10
paddingLeft	50
paddingRight	50
paddingTop	10

.navigationlevel1

The `.navigationlevel1` class specifies properties for the first level navigation heading.

Property	Default value
backgroundAlpha	0
backgroundColor	#FFFFFF
borderStyle	none
color	#0b333c
fillColors	#336699
fontSize	14
fontWeight	bold
rollOverColor	#006699
selectionColor	#FFFFFF
textDecoration	normal
textRollOverColor	#333333
textSelectedColor	#000000
themeColor	#006699

.navigationlevel2

The `.navigationlevel2` class specifies properties for the second level navigation heading.

Property	Default value
<code>alpha</code>	1.0
<code>backgroundColor</code>	#006699
<code>color</code>	#FFFFFF
<code>fillColors</code>	#3399CC, #3399CC
<code>fontSize</code>	12
<code>fontWeight</code>	normal
<code>rollOverColor</code>	#006699
<code>selectionColor</code>	#006699
<code>textDecoration</code>	normal
<code>textRollOverColor</code>	#333333
<code>textSelectedColor</code>	#000000
<code>themeColor</code>	#006699

.navigationlevel3

The `.navigationlevel3` class specifies properties for the third level navigation heading.

Property	Default value
<code>backgroundColor</code>	#FFFFFF
<code>color</code>	#0b333c
<code>fillColors</code>	#FFFFFF, #FFFFFF
<code>fontSize</code>	14
<code>fontWeight</code>	bold
<code>rollOverColor</code>	#FFFFFF
<code>selectionColor</code>	#FFFFFF
<code>textDecoration</code>	normal
<code>textRollOverColor</code>	#888888
<code>textSelectedColor</code>	#FFFFFF
<code>themeColor</code>	#FFFFFF

.navigationlevel4

The `.navigationlevel4` class specifies properties for the fourth level navigation heading.

Property	Default value
<code>backgroundColor</code>	#FFFFFF
<code>color</code>	#0b333c
<code>fillColors</code>	#FFFFFF, #FFFFFF
<code>fontSize</code>	14
<code>fontWeight</code>	bold
<code>rollOverColor</code>	#FFFFFF
<code>selectionColor</code>	#FFFFFF
<code>textDecoration</code>	normal
<code>textRollOverColor</code>	#888888
<code>textSelectedColor</code>	#FFFFFF
<code>themeColor</code>	#FFFFFF

.navigationlevel5

The `.navigationlevel5` class specifies properties for the fifth level navigation heading.

Property	Default value
<code>backgroundColor</code>	#FFFFFF
<code>color</code>	#0b333c
<code>fillColors</code>	#FFFFFF, #FFFFFF
<code>fontSize</code>	9
<code>fontWeight</code>	normal
<code>rollOverColor</code>	#FFFFFF
<code>selectionColor</code>	#FFFFFF
<code>textDecoration</code>	normal
<code>themeColor</code>	#FFFFFF

.navigationOver

The `.navigationOver` class specifies the border color that is used when the form filler moves the mouse pointer over a navigation control.

Property	Default value
fillColors	#3366CC, #3366CC

.navigationSelected

The `.navigationSelected` class specifies the background color that is used when the form filler selects a navigation control.

Property	Default value
fillColors	#3366CC, #3366CC

.panelcaption

The `.panelcaption` class defines the text attributes of field captions in a panel.

Property	Default value
color	#333333
fontSize	12
fontStyle	normal
fontWeight	normal
textDecoration	normal

.paneldata

The `.paneldata` class defines the properties for the area of a panel layout that displays form objects and data.

Property	Default value
backgroundAlpha	1.0
backgroundColor	#FFFFFF
borderAlpha	1.0
borderColor	#336699
color	#FFFFFF
cornerRadius	10
headerAlphas	1.0,1.0

Property	Default value
headerColors	#336699, #336699
roundedBottomCorners	true

.panelhelp

The `.panelhelp` class specifies a background color for the help that appears in the data entry panel.

Property	Default value
alpha	1.0
backgroundAlpha	1.0
backgroundColor	#999999
borderColor	#336699
borderStyle	solid
color	#000000
cornerRadius	5
headerAlphas	1.0, 1.0
headerColors	#336699, #336699

.panelitem

The `.panelitem` class specifies style properties for form guide fields.

Property	Default value
backgroundAlpha	1.0
backgroundColor	#CCCCCC
borderStyle	inset
color	#333333
dropShadowEnabled	false
fontSize	12
fontStyle	normal
fontWeight	normal
textDecoration	normal
themeColor	#336699

.panelnav

The `.panelnav` class defines the colors for the controls in the navigation panel, such as accordions or tabs.

Property	Default value
backgroundAlpha	1.0
backgroundColor	#FFFFFF
barColor	#3366cc
borderAlpha	1.0
borderColor	#336699
color	#FFFFFF
cornerRadius	10
headerAlphas	1.0, 1.0
headerColors	#336699, #336699
roundedBottomCorners	true

.paneltext

The `.paneltext` class defines text attributes of the panel text.

Property	Default value
color	#000000
fontSize	12
fontStyle	normal
fontWeight	normal
textDecoration	normal

.progressbar

The `.progressbar` class defines the background colors of the progress bar.

Property	Default value
backgroundColor	#336699
barColor	#336699

.repeater

The `.repeater` class defines the background colors of the repeater components.

Property	Default value
backgroundColor	#336699
themeColor	#336699

.videocontrol

The `.videocontrol` class controls the color of the background and of text.

Property	Default value
backgroundColor	#AFAFAF
color	#000000

ComboBox

The `ComboBox` class specifies the default styles for all drop-down lists in a form guide.

Property	Default value
color	#000000
dropShadowEnabled	true
fontFamily	Myriad Pro
fontSize	11

DateField

The `DateField` class specifies the default formatting for all date fields in a form guide.

Property	Default value
color	#000000
dropShadowEnabled	true
fontFamily	Myriad Pro
fontSize	12

GAIcon

The `GAIcon` class specifies the format for the logo in a form guide.

Property	Default value
iconColors	#336699, #336699
iconTextColor	#FFFFFF

Label

The `Label` class specifies the format for static text in a form guide.

Property	Default value
color	#000000

RadioButton

The `RadioButton` class specifies formatting for all radio buttons used in a form guide.

Property	Default value
dropShadowEnabled	true
fontFamily	Myriad Pro
fontSize	12

TextInput

The `TextInput` class specifies the format for all text fields used in a form guide.

Property	Default value
color	#000000
dropShadowEnabled	true
fontFamily	Myriad Pro
fontSize	12

TextInputMask

The `TextInputMask` class specifies the style properties of the mask input comb field. It enables guided data entry by providing visual cues to the user filling the form guide. For example, a postal code field would display as a field with 6 cells.

Property	Default value
backgroundColor	#333333
cellColor	#CCCCCC

TextInputSymbol

The `TextInputSymbol` class specifies the format for currency symbols used in a form guide.

Property	Default value
prefixBackgroundColor	#333333
prefixColor	#CCCCCC
suffixBackgroundColor	#333333
suffixColor	#CCCCCC

Form guide CSS properties

alpha

The `alpha` property specifies the transparency of the color of text on a button.

Syntax

`alpha=" [0.0..1.0] "`

Applies to

- [.buttons](#)
- [.guidehelp](#)
- [.navigationlevel2](#)
- [.panelhelp](#)

backgroundAlpha

The backgroundAlpha property specifies the transparency of the background color.

Syntax

```
backgroundAlpha=" [0.0..1.0] "
```

Applies to

- [.application](#)
- [.buttons](#)
- [.guide](#)
- [.guidehelp](#)
- [.navigationlevel1](#)
- [.paneldata](#)
- [.panelhelp](#)
- [.panelitem](#)
- [.panelnav](#)
- [TextInputMask](#)

backgroundColor

The backgroundColor property specifies the guide background color.

Syntax

```
backgroundColor="color-name | color-rgb | color-hex | transparent"
```

Applies to

- [.application](#)
- [.fieldhelp](#)
- [.guide](#)
- [.guidehelp](#)
- [.navigationlevel1](#)
- [.navigationlevel2](#)
- [.navigationlevel3](#)
- [.navigationlevel4](#)
- [.navigationlevel5](#)
- [.paneldata](#)
- [.panelhelp](#)
- [.panelitem](#)
- [.panelnav](#)
- [.progressbar](#)
- [.repeater](#)

- [.videocontrol](#)
- [TextInputMask](#)

backgroundGradientColors

The `backgroundGradientColors` property specifies the two color extremes of the background color gradient.

Syntax

```
backgroundGradientColors=" [color-name | color-rgb | color-hex | transparent],  
[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[.application](#)

barColor

The `barColor` property specifies the color of the progress indicator.

Syntax

```
barColor=" [color-name | color-rgb | color-hex | transparent],  
[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[.guide](#)
[.panelnav](#)
[.progressbar](#)

basewrapper

The `basewrapper` property specifies the name of the form guide layout.

Syntax

```
basewrapper=" [form guide layout name]"
```

Applies to

[.application](#)

borderAlpha

The borderAlpha property specifies the standard Flex style for a panel.

Syntax

```
borderAlpha="0.0..1.0"
```

Applies to

[.guide](#)

[.guidehelp](#)

[.paneldata](#)

[.panelnav](#)

borderColor

The borderColor property specifies the color of the border around field help.

Syntax

```
borderColor="color-name | color-rgb | color-hex | transparent"
```

Applies to

[.fieldhelp](#)

[.guide](#)

[.guidehelp](#)

[.paneldata](#)

[.panelhelp](#)

[.panelnav](#)

borderStyle

The borderStyle property specifies the required default border value used by the supplied layouts.

Syntax

```
borderStyle=""
```

Applies to

[.guide](#)

[.navigationlevel1](#)

[.panelhelp](#)

[.panelitem](#)

cellColor

The `cellColor` property specifies the style attribute for the `TextInputMask`.

Syntax

```
cellColor="[color-name | color-rgb | color-hex | transparent],  
[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[TextInputMask](#)

color

The `color` property specifies the color to use for text in form guide help.

Syntax

```
color="[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[.buttons](#)

[.fieldhelp](#)

[.guidehelp](#)

[.navigationlevel1](#)

[.navigationlevel2](#)

[.navigationlevel3](#)

[.navigationlevel4](#)

[.navigationlevel5](#)

[.panelcaption](#)

[.paneldata](#)

[.panelhelp](#)

[.panelitem](#)

[.panelnav](#)

[.paneltxt](#)

[.videocontrol](#)

[ComboBox](#)

[TextField](#)

[Label](#)

[TextInput](#)

cornerRadius

The `cornerRadius` property specifies the roundness of the panel border.

Syntax

`cornerRadius=" [0..80] "`

Applies to

- [.guide](#)
- [.guidehelp](#)
- [.paneldata](#)
- [.panelhelp](#)
- [.panelnav](#)

dropShadowEnabled

The `dropShadowEnabled` property specifies whether to add a drop shadow effect to field help.

Syntax

`dropShadowEnabled="true | false"`

Applies to

[.fieldhelp](#)
[.panelitem](#)
[ComboBox](#)
[DateField](#)
[RadioButton](#)
[TextInput](#)

fillAlphas

The `fillAlphas` property specifies the transparency of the gradient colors defined by the `fillColors` property.

Syntax

`fillAlphas=" [0.0..1.0] , [0.0..1.0] "`

Applies to

[.buttons](#)

fillColors

The `fillColors` property specifies the two color extremes of the color gradient to use for button color.

Syntax

```
fillColors="[color-name | color-rgb | color-hex | transparent],  
[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[.buttons](#)
[.navigationlevel1](#)
[.navigationlevel2](#)
[.navigationlevel3](#)
[.navigationlevel4](#)
[.navigationlevel5](#)
[.navigationOver](#)
[.navigationSelected](#)

fontFamily

The `fontFamily` property specifies the font typeface.

Syntax

```
fontFamily="[font name]"
```

Applies to

[ComboBox](#)
[DateField](#)
[RadioButton](#)
[TextInput](#)

fontSize

The `fontSize` property specifies the font size.

Syntax

`fontSize="integer"`

Applies to

[.fieldhelp](#)

[.navigationlevel1](#)

[.navigationlevel2](#)

[.navigationlevel3](#)

[.navigationlevel4](#)

[.navigationlevel5](#)

[.panelcaption](#)

[.panelitem](#)

[.paneltxt](#)

[ComboBox](#)

[TextField](#)

[RadioButton](#)

[TextInput](#)

fontStyle

The `fontStyle` property specifies whether to use italicized font.

Syntax

`fontStyle="normal | italic"`

Applies to

[.fieldhelp](#)

[.panelcaption](#)

[.panelitem](#)

[.paneltxt](#)

fontWeight

The `fontWeight` property specifies whether to use font bolding.

Syntax

```
fontWeight="normal | bold"
```

Applies to

[.fieldhelp](#)

[.navigationlevel1](#)

[.navigationlevel2](#)

[.navigationlevel3](#)

[.navigationlevel4](#)

[.navigationlevel5](#)

[.panelcaption](#)

[.panelitem](#)

[.panelttext](#)

gradientColors

The `gradientColors` property specifies two colors for the form guide container. Guide Builder uses the two colors to create a gradient fill.

Syntax

```
gradientColors=" [color-name | color-rgb | color-hex | transparent] ,  
               [color-name | color-rgb | color-hex | transparent]"
```

Applies to

[.fieldhelp](#)

headerAlphas

The `headerAlphas` property controls the transparency of the title area.

Syntax

`headerAlphas= " "`

Applies to

[.guidehelp](#)

[.paneldata](#)

[.panelhelp](#)

[.panelnav](#)

headerColors

The `headerColors` property specifies the color of the guide help title area.

Syntax

`headerColors= "[color-name | color-rgb | color-hex | transparent],
[color-name | color-rgb | color-hex | transparent]"`

Applies to

[.guidehelp](#)

[.paneldata](#)

[.panelhelp](#)

[.panelnav](#)

headerHeight

The `headerHeight` property specifies title bar size for panel areas (auto defined).

Syntax

`headerHeight= "[integer]"`

Applies to

[.guide](#)

horizontalAlign

The `horizontalAlign` property specifies the standard Flex style for most containers.

Syntax

```
horizontalAlign=""
```

Applies to

[.navigationbase](#)

iconColors

The `iconColors` property specifies a pair of fill colors for the icon component. The icon component consists of a sphere with a gradient fill and the letter "i" in the middle. The icon component is used for the information icon.

Syntax

```
iconColors="[color-name | color-rgb | color-hex | transparent],  
[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[GIcon](#)

iconTextColor

The `iconTextColor` property specifies the color of the text inside the gradient sphere of the icon component.

Syntax

```
iconTextColor="[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[GIcon](#)

paddingBottom

The `paddingBottom` property specifies the border around the bottom edge of the form guide.

Syntax

`paddingBottom=" [0..80] "`

Applies to

[.application](#)

[.guide](#)

[.layoutobjects](#)

[.layoutrepeaterobjects](#)

[.logo](#)

[.navigationbase](#)

paddingLeft

The `paddingLeft` property specifies the border around the left edge of the form guide.

Syntax

`paddingLeft=" [0..80] "`

Applies to

[.application](#)

[.guide](#)

[.layoutobjects](#)

[.layoutrepeaterobjects](#)

[.navigationbase](#)

paddingRight

The `paddingRight` property specifies the border around the right edge of the form guide.

Syntax

`paddingRight=" [0..80] "`

Applies to

[.application](#)

[.guide](#)

[.layoutobjects](#)

[.layoutrepeaterobjects](#)

[.logo](#)

[.navigationbase](#)

paddingTop

The paddingTop property specifies the border around the top edge of the form guide.

Syntax

paddingTop=" [0..80] "

Applies to

[.application](#)

[.guide](#)

[.layoutobjects](#)

[.layoutrepeaterobjects](#)

[.navigationbase](#)

prefixBackgroundColor

The prefixBackgroundColor property specifies the background color of the prefix area of the TextInputSymbol component.

Syntax

prefixBackgroundColor=" [color-name | color-rgb | color-hex | transparent] "

Applies to

[TextInputSymbol](#)

prefixColor

The prefixColor property specifies the color of the text inside the prefix area of the TextInputSymbol component.

Syntax

prefixColor=" [color-name | color-rgb | color-hex | transparent] "

Applies to

[TextInputSymbol](#)

rollOverColor

The `rollOverColor` property specifies the border color that is used when the form filler moves the mouse pointer over a navigation control.

Syntax

```
rollOverColor=" [color-name | color-rgb | color-hex | transparent] "
```

Applies to

[.navigationlevel1](#)

[.navigationlevel2](#)

[.navigationlevel3](#)

[.navigationlevel4](#)

[.navigationlevel5](#)

roundedBottomCorners

The `roundedBottomCorners` property specifies whether the panel border is rounded.

Syntax

```
roundedBottomCorners="true | false"
```

Applies to

[.guidehelp](#)

[.paneldata](#)

[.panelnav](#)

selectionColor

The `selectionColor` property specifies the background color that is used when the form filler selects a navigation control.

Syntax

```
selectionColor=" [color-name | color-rgb | color-hex | transparent] "
```

Applies to

[.navigationlevel1](#)

[.navigationlevel2](#)

[.navigationlevel3](#)

[.navigationlevel4](#)

[.navigationlevel5](#)

suffixBackgroundColor

The `suffixBackgroundColor` property specifies the background color of the suffix area of the `TextInputSymbol` component.

Syntax

```
suffixBackgroundColor = "[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[TextInputSymbol](#)

suffixColor

The `suffixColor` property specifies the color of the text inside the suffix area of the `TextInputSymbol` component.

Syntax

```
suffixColor = "[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[TextInputSymbol](#)

textDecoration

The `textDecoration` property specifies whether to use font underlining.

Syntax

```
textDecoration = "normal | underline"
```

Applies to

[.fieldhelp](#)

[.guidehelp](#)

[.navigationlevel1](#)

[.navigationlevel2](#)

[.navigationlevel3](#)

[.navigationlevel4](#)

[.navigationlevel5](#)

[.panelcaption](#)

[.panelitem](#)

[.paneltextright](#)

textRollOverColor

The `textRollOverColor` property specifies the text color that is used when a form filler moves the mouse pointer moves over a navigation control.

Syntax

```
textRollOverColor="[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[.navigationlevel1](#)
[.navigationlevel2](#)
[.navigationlevel3](#)
[.navigationlevel4](#)

textSelectedColor

The `textSelectedColor` property specifies the text color of an element of a component when the element is selected.

Syntax

```
textSelectedColor="[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[.navigationlevel1](#)
[.navigationlevel2](#)
[.navigationlevel3](#)
[.navigationlevel4](#)

themeColor

The `themeColor` property specifies the theme color to use for buttons on a form guide.

Syntax

```
themeColor="[color-name | color-rgb | color-hex | transparent],  
[color-name | color-rgb | color-hex | transparent]"
```

Applies to

[.buttons](#)
[.navigationlevel1](#)
[.navigationlevel2](#)
[.navigationlevel3](#)

[.navigationlevel4](#)

[.navigationlevel5](#)

[.panelitem](#)

[.repeater](#)

version

The `version` property specifies the version of the CSS file.

Note: This property is for information purposes only.

Syntax

`version=" [version number] "`

Applies to

[.application](#)

Index

A

AccordionNav component 22
ActionScript
 control, creating 17
 examples 18
AddPanelButton component 21

B

building Flex library projects 8
Button Bar form guide layout 12
button components 21

C

components, custom 20
controls, creating 17
CopyPanelButton component 21
creating
 controls 17
 Flex library projects 7
 form guide layouts 11
 panel layouts 14

D

documentation resources 4

E

examples
 form guide layout 12
 navigation tree 18
 Next panel button 18
 panel layout structure 14

F

Flex Builder 7
Flex library projects
 building 8
 creating 7
 folder structure 8
folder structure, Flex library project 8
form guide layouts
 about 10
 creating 11
 example 12
form guides
 creating Flex library projects for 7

G

Guide Builder, making SWC files available to 8

H

HelpBox component 21
HelpPanel component 21
HelpVideo component 21

L

label components 22

M

MXML control, creating 17
MxTreeNav component 22

N

navigation components 22
navigation tree, example 18
Next panel button 18
NextPanelButton component 21

O

Outline view, panel layout 15
output components 22

P

panel layouts
 creating 14
 Outline view, example 15
 structure 14
PanelContent output component 22
PanelTitle label component 22
PDFBox output component 22
PreviousPanelButton component 21
ProgressBar output component 22
ProgressSectionBarNav component 22

R

RemovePanelButton component 21
resources, documentation 4

S

SectionTitle label component 22
SubmitButton component 21
SWC files
 adding to a Flex library project 7
 importing into Guide Builder 8

T

TabTabNav component 22
ToolBar output component 22